ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Noval Architecture of UTMI Module in USB

A.Trinadha Rao
Faculty,
Department of ECE,
Raghu Engineering College,
Visakhapatnam, A.P, India.

CH.Ganesh
Faculty,
Department of ECE,
Raghu Engineering College,
Visakhapatnam, A.P, India.

M.Sravan Kumar
B.Tech (Pursuing),
Department of ECE,
Raghu Engineering College,
Visakhapatnam, A.P, India.

S.V.Vinay Kumar
B.Tech (Pursuing),
Department of ECE,
Raghu Engineering College,
Visakhapatnam, A.P, India.

P.Lavanya
B.Tech (Pursuing),
Department of ECE,
Raghu Engineering College,
Visakhapatnam, A.P, India.

Abstract:

The universal serial bus (USB) transceiver macro cell interface (UTMI) is a two wire, bi-directional serial bus interface. UTMI consists of transmitting and receiving sections, in which the transmitter of the UTMI sends data to different USB devices through D+ and D- lines whereas the receiver gets data on the same lines. UTMI implementations depends on transmission rates, those Low Speed (1.5MHz), Full Speed (12MHz) and High Speed (480MHz). UTMI is one of the important functional blocks of USB controller and three functional blocks in USB controller those are Serial Interface Engine (SIE), UTMI and Device Specific Logic (DSL). The parallel data from SIE is taken into the transmit hold register and it's sent to transmit shift register from where the data is converted serially. This serial data is bit stuffed to perform data transitions for clock recovery and NRZI (1) encoding. Then the encoded data is sent on to the serial bus. When the data is received on the serial bus, it is decoded, bit unstuffed and is sent to receive shift register. After the shift register is full, the Data is sent to receive hold register. This presentation reveals the FPGA implementation of UTMI transmission rate providing with USB 2.0 specifications. Further UTMI has been designed by using VHDL code and simulated, synthesized and programmed to the targeted Spartan family of FPGA in the Xilinx environment.

Index Terms:

USB Transceiver Macro Cell, Serial Interface Engine, Device Specific Logic, Bit Stuff, Bit Un Stuff, Encoder, Decoder

I.INTRODUCTION:

The fig-1 shows the block diagram of USB controller, which is present in every USB device. There are three major functional blocks in a USB 2.0 controller. The USB 2.0 Transceiver Microcell Interface (UTMI), the Serial Interface Engine (SIE), and the device specific logic.

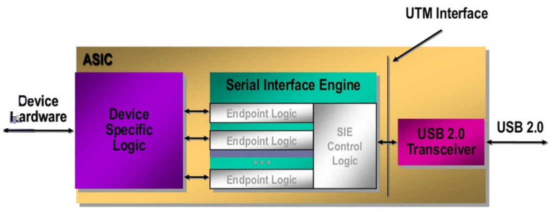


Fig1-Block diagram of USB controller

A.UTMI:

This block handles the low level USB protocol and signalling. This includes features such as; data serialization and deserialization, bit stuffing and clock recovery and synchronization. The primary focus of this block is to shift the clock domain of the data from the USB 2.0 rate to one that is compatible with the general logic in the ASIC. The UTMI is designed to support HS/FS, FS Only and LS Only UTM implementations. The three options allow a single SIE implementation to be used with any speed USB transceiver. A vendor can choose the transceiver performance that best meets their needs. A HS/FS implementation of the transceiver can operate at either a 480 Mb/s or a 12 Mb/s rate. Two modes of operation are required to properly emulate High-speed device connection and suspend/resume features of USB 2.0, as well as Full-speed connections if implementing a Dual-Mode device FS Only and LS Only UTM implementations do not require the speed selection signals since there is no alternate speed to switch to Serial Interface Engine.

B. SERIAL INTERFACE ENGINE:

This block can be further sub-divided into 2 types of sub-blocks; the SIE Control Logic and the Endpoint logic. The SIE Control Logic contains the USB PID and address recognition logic, and other sequencing and state machine logic to handle USB packets and transactions. The Endpoint Logic contains the endpoint specific logic: endpoint number recognition, FIFOs and FIFO control, etc. Generally the SIE Control Logic is required for any USB implementation while the number and types of endpoints will vary as function of application and performance requirements. SIE logic module can be developed by peripheral vendors or purchased from IP vendors. The standardization of the UTMI allows compatible SIE VHDL to drop into an ASIC that provides the macro cell.

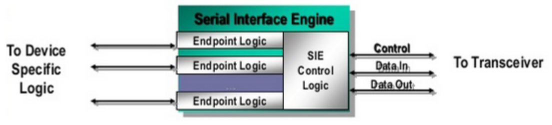


Fig2. Block diagram of SIE

C. DEVICE SPECIFIC LOGIC:

This is the glue that ties the USB interface to the specific application of the device.

II. USB TRANCIEVER MACROCELL INTERFACE (UTMI):

A. Introduction: Universal Serial Bus (USB) Transceiver Macro cell Interface (UTMI) is one of the most important blocks of USB Controller. This block handles the low level USB protocol and signalling. This includes features such as data serialization, de serialization, bit stuffing, bit de stuffing, Non Return to Zero Invert on „1"(NRZI) encoding, decoding, clock recovery and synchronization. The primary focus of this block is to shift the clock domain of the data from the USB 2.0 rate to one that this compatible with the general logic in the ASIC.

B. The Transmitter Module:

The diagram of the UTMI transmitter is shown in Figure3. The transmitter module has been enforced by considering the subsequent specifications. The synchronise pattern "01111110" should be transmitted like a shot once the transmitter is initiated by the SIE. once six consecutive „1"s occur within the knowledge stream a zero to be inserted. the information ought to be encoded victimization Non come to Zero Invert on one (NRZI -1) cryptography technique. The EOP pattern 2 single over zeroes(D+ and D-lines ar carrying zero for 2 clock cycles) and somewhat one ought to be transmitted once every packet or once SIE suspends the transmitter The transmitter logic facilitates synchronise transmission, holding parallel 8- bit knowledge from SIE, parallel to serial conversion of information, bit stuffing, NRZI cryptography , transmission of the information and EOP transmission on to the serial bus.

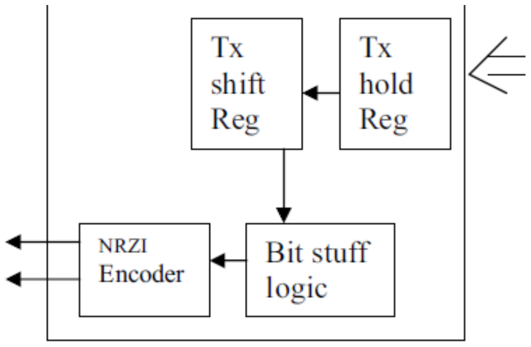


Fig3. Block diagram of UTMI Transmitter

C. The Receiver Module:

The diagram of the UTMI receiver is shown in Figure4. The receiver module has been enforced by considering the subsequent specifications. When correct pattern is detected that ought to be intimated to the SIE. If a zero isn't detected when six consecutive „1"s a slip-up ought to be reported to the SIE. When EOP pattern is detected that ought to be intimated to the SIE. The receiver logic facilitates correct detection, NRZI decryption, bit unstuffing, serial to parallel conversion of knowledge, receive error reportage and EOP detection.

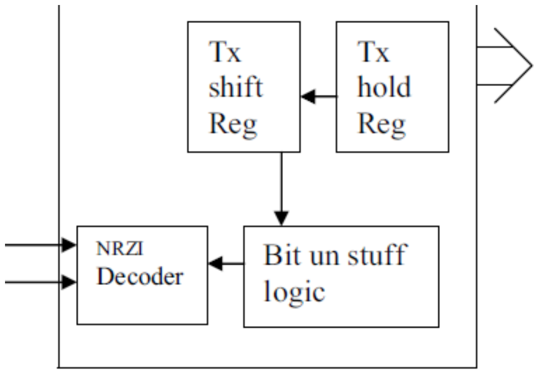


Fig4. Block diagram of UTMI Receiver

D. The Transceiver Module:

Figure 5 and 6 shows, the transmitter and the receiver modules combined as transceiver (UTMI) module. This transceiver met all the USB3.0 specifications thought-about on top of. The transceiver logic facilitates the output of the transmitter to feed to the input of the receiver for purposeful verification. The Transceiver module has been designed with the issues of individual modules of the transmitter and therefore the receiver Specifications.

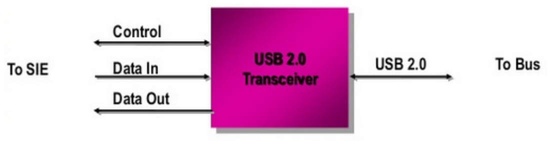


Fig5. Block Diagram of USB

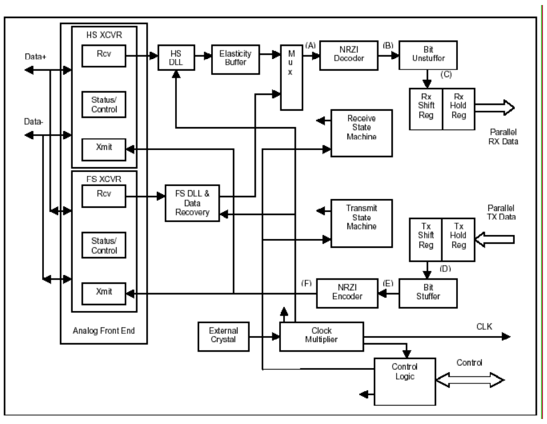


Fig6.Functional Block Diagram of UTMI

III. DESIGN ASPECTS OF NEW UTMI MODULE:

The present UTMI has been designed consistent with the subsequent specifications provided by the USB 2.0 protocol. correct and finish of Packet (EOP) generation by the transmitter. Correct and EOP detection by the receiver receive error reportage. enabling or disabling the bit stuffer and NRZI encoder depends on the operational mode. Suspension of the transceiver by the SIE. Further the UTMI is split into 2 necessary modules that are the Transmitter module and also the Receiver module. during this section the look issues of those modules are explained individually and integrated to urge prime level Transceiver (UTMI) module. The UTMI is divided into mainly two modules, which are the transmitter module and the receiver module. Modified UTMI functional diagram is shown in fig7.

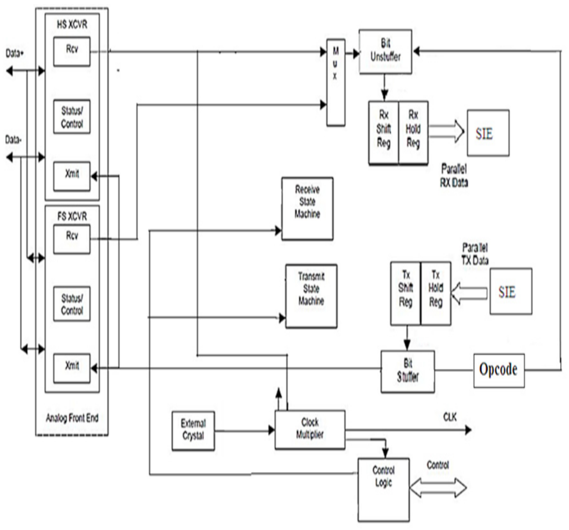
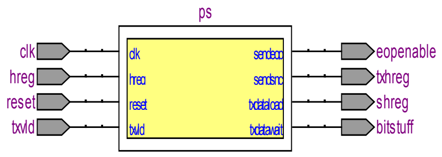


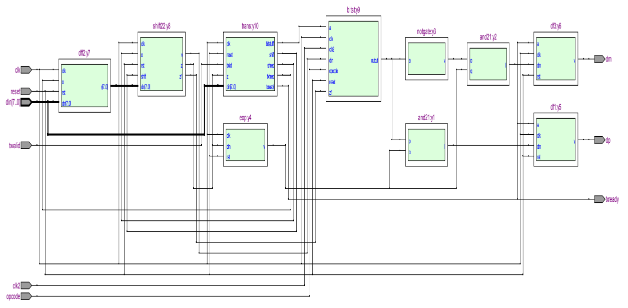
Fig7. Modified functional Block Diagram of UTMI

TRANSMITTER SECTION:

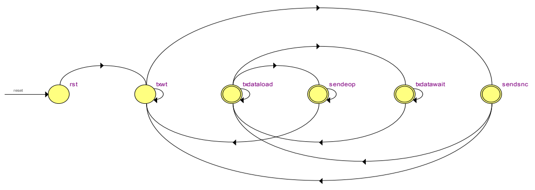
RTL VIEW OF TX CONTROLLER:



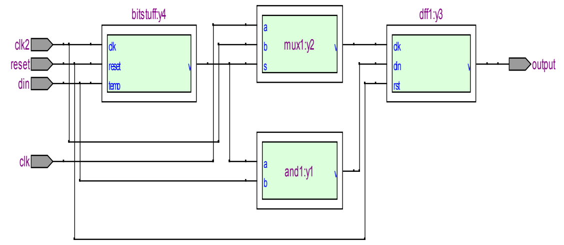
RTL SCHEMATIC VIEW OF TRANSMITTER MODULE:



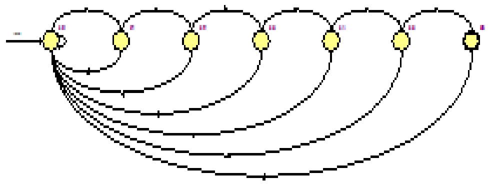
STATE DIAGRAM OF TRANSMITTER MODULE:



RTL SCHEMATIC VIEW OF BIT STUFF MODULE:

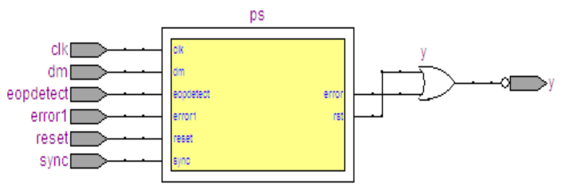


STATE DIAGRAM OF BIT STUFF MODULE:

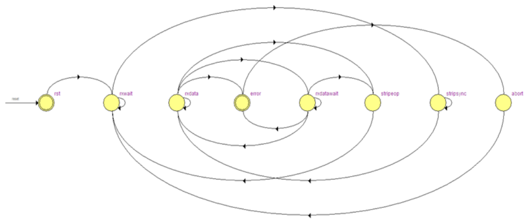


RECEIVER SECTION:

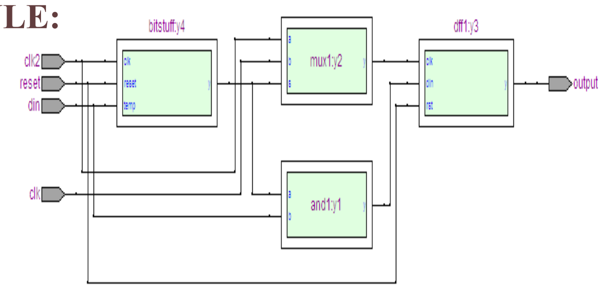
RTL VIEW OF RECEIVER CONTROLLER:



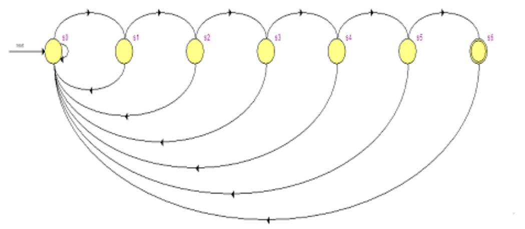
STATE DIAGRAM FOR RECEIVER:



RTL VIEW OF BIT UNSTUFFING MODULE:



STATE DIAGRAM FOR BIT UNSTUFFING:



IV. RESULTS:

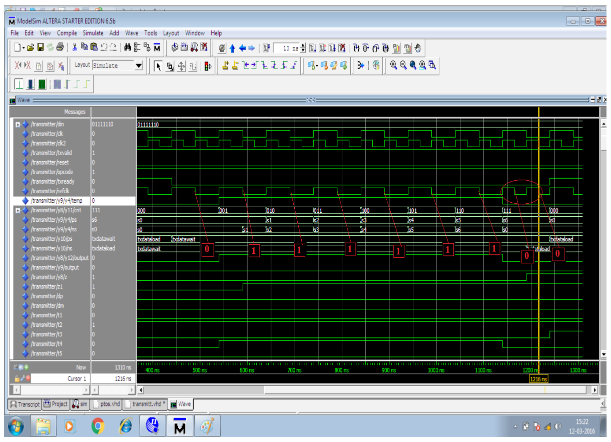
The individual modules of the UTMI are designed using VHDL as stated above and they are simulated within the Xilinx based Model Sim 6.0 environment.

TRANSMITTER SECTION:

A. The Transmitter Module:

The Figure shows the Simulation results of UTMI transmitter.

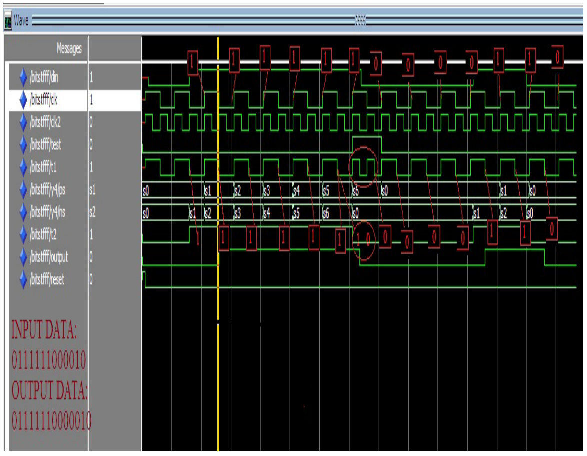
SIMULATION RESULT OF TRANSMITTER:


ModelSim ALTERA STARTER EDITION 6.5b
File Edit View Compile Simulate Add Wave Tools Layout Window Help
Layout Simulate
Wave
Messages
Transcript
Project
sim
ptos.vhd
transmitt.vhd
Wave

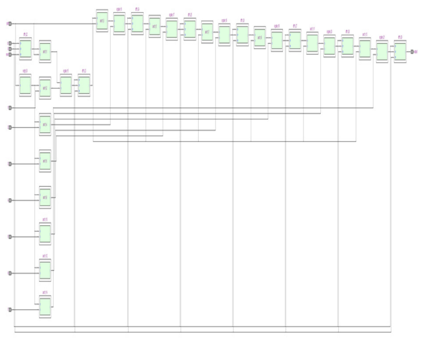
/transmitter/din 01111110
/transmitter/clk 0
/transmitter/clk2 0
/transmitter/txvalid 1
/transmitter/reset 0
/transmitter/opcode 1
/transmitter/txready 0
/transmitter/refclk 0
/transmitter/y9/y4/temp 0
/transmitter/y8/y11/cnt 111
/transmitter/y9/y4/ps s6
/transmitter/y9/y4/ns s0
/transmitter/y10/ps txdatawait
/transmitter/y10/ns txdataload
/transmitter/y8/y12/output 0
/transmitter/y9/output 0
/transmitter/y8/z 1
/transmitter/z1 1
/transmitter/dp 0
/transmitter/dm 0
/transmitter/t1 0
/transmitter/t2 1
/transmitter/t3 0
/transmitter/t4 0
/transmitter/t5 0

Now 1310 ns
Cursor 1 1216 ns

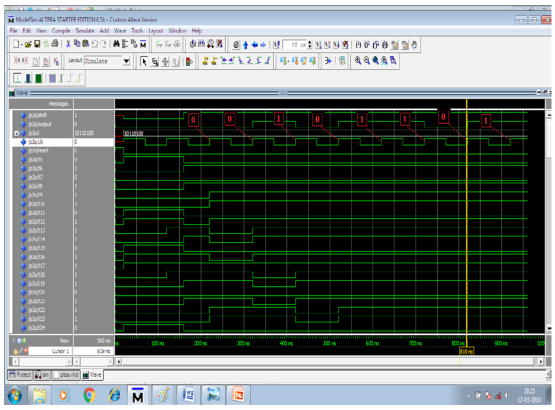
SIMULATION RESULT OF BIT STUFF MODULE:



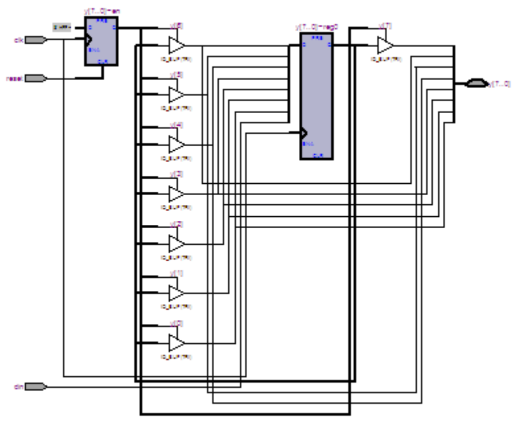
RTL VIEW OF PARALLEL TO SERIEL



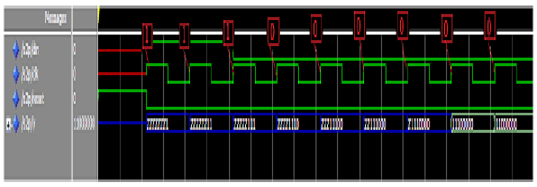
SIMULATION RESULT OF PARALLEL TO SERIEL:



RTL VIEW OF SERIEL TO PARALLEL



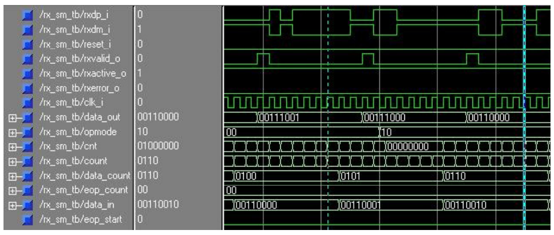
SIMULATION RESULT OF SERIEL TO PARALLEL



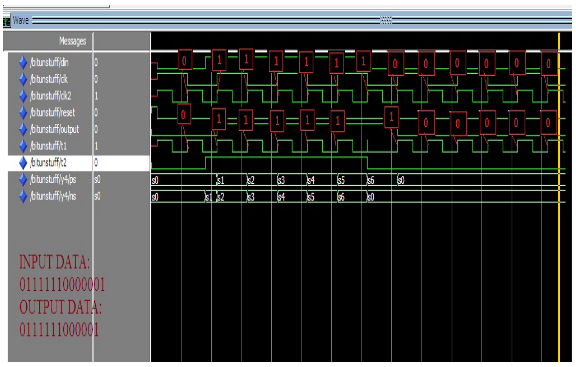
RECEIVER SECTION:

B. The Receiver Module:

The Figure shows the Simulation results of UTMI receiver. once correct is detected rxactive is declared. the info gift on rxdp, rxdm lines is decoded, serial to parallel born-again and sent to the SIE through knowledge out bus by declarative rxvalid signal.



SIMULATION RESULT OF BIT UNSTUFFING:



VII. CONCLUSION:

From the above results it can be concluded that the individual modules of UTMI have been designed, verified functionally using VHDL simulator. The UTMI Transmitter is capable of converting parallel data into serial bits, performing bit stuffing. The UTMI Receiver is capable of performing bit unstuffing and converting serial bits into parallel data. The functional simulation has been successfully carried out. UTMI has been design by using VHDL code and simulate using Xilinx, model-sim. This module is implement by using quartus-II tools.

FUTURE SCOPE:

The UTMI has been implemented is 8-bit one, it can also be extended to 16- bit UTMI. It can also be designed to generate CRCs for control and data packets. If an SIE and Device specific logic are designed, the combination of UTMI, SIE and Device specific logic can be used as controller of a USB device.

ACKNOWLEDGMENT:

We express my deep sense of gratitude and thanks Chairman sir Sri. Raghu Kalidindi, Dean Prof S. Veerabhadraiah, Head of the Department Mr. K.Phaninder Vinay and Principal Dr. R.Kameswara Rao sir of Raghu Engineering College Visakhapatnam, India for giving us this opportunity to carry out my project work at highly esteemed Organization.

REFERENCES:

[1]. USB 2.0 Specification, April 27, 2000
[2]. USB 2.0 Transceiver Macrocell Interface (UTMI) Specification, version 1.05, March 29, 2001
[3]. On-The-Go Supplement to the USB 2.0 Specification, revision 1.0, Dec 18, 2001
[4]. UTMI+ Specification, revision 0.9, February 21, 2003
[5]. VHDL With Example Douglas .L .Perry
[6]. Data and Computer Communications by William Stallings
[7]. Computer Networks by Andrew S. Tannenbaum
[8]. www.opencore.org
[9]. www.digitalcoredesign.org